**Q3: Implementing an RNN for Text Generation**

def generate\_text(model, start\_string, length=500, temperature=1.0):

    # Filter out characters not found in char\_to\_idx

    start\_string = ''.join([char for char in start\_string if char in char\_to\_idx])

    # If the start string is empty after filtering, return an error message

    if not start\_string:

        return "Error: None of the characters in the start string are valid."

    input\_sequence = [char\_to\_idx[char] for char in start\_string]

    input\_sequence = np.expand\_dims(input\_sequence, axis=0)  # Shape: (1, sequence\_length)

    generated\_text = start\_string

    for \_ in range(length):

        predictions = model.predict(input\_sequence, verbose=0)  # Predict the next character probabilities

        predictions = predictions[0, :]  # Get prediction for the last character in the sequence (2D array)

        # Apply temperature scaling

        predictions = predictions / temperature  # Scale the predictions

        predictions = np.exp(predictions) / np.sum(np.exp(predictions))  # Softmax (normalize)

        # Sample the next character based on the probabilities

        next\_char\_idx = np.random.choice(len(chars), p=predictions)

        next\_char = idx\_to\_char[next\_char\_idx]

        generated\_text += next\_char

        # Update input sequence (shift the window)

        input\_sequence = np.append(input\_sequence, [[next\_char\_idx]], axis=1)

        input\_sequence = input\_sequence[:, 1:]  # Remove the first element to maintain sequence length

    return generated\_text

# Test with a valid start string and different temperature values

start\_string = "shall i compare thee to a summer's day"

# Testing with different temperature values for more controlled output

generated\_text\_0\_5 = generate\_text(model, start\_string, length=500, temperature=0.5)

generated\_text\_0\_7 = generate\_text(model, start\_string, length=500, temperature=0.7)

generated\_text\_1\_0 = generate\_text(model, start\_string, length=500, temperature=1.0)

# Print the generated texts for comparison

print("Generated Text (Temperature 0.5):", generated\_text\_0\_5)

print("Generated Text (Temperature 0.7):", generated\_text\_0\_7)

print("Generated Text (Temperature 1.0):", generated\_text\_1\_0)